Modelling and optimize your data warehouse

Christian Henrik Reich

ChristianHenrikReich@gmail.com

Premium Sponsors

Raffle Prizes

Standard Sponsors

TIMEXTENDER

[stoltze][it]

ORANGEMAN

exmon

KPMG

Fellowwind

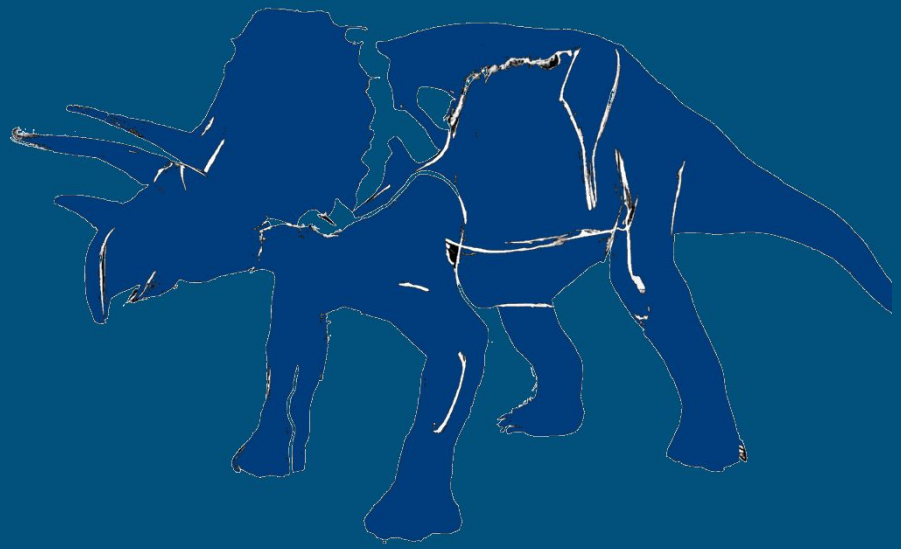
Christian Henrik Reich

- Working professionally since 1999, building everything from embedded computer games in set top boxes to analytics platforms
- Works as a architect, developer and advisor @ Kapacity
- Part of the company Mugato.com (#IRM)
- In the Øredev program committee, part of the data track and the off track
- Teaching data engineering and AI at Redi School
- Former assessment test author for Pluralsight
- Currently holds 23 Microsoft Certifications

What are going to cover

- Is data modelling and patterns still relevant in 2023
- Modelling
- Implementation
- Anti-patterns
- Why it works

Is data modelling and patterns still relevant in these modern times?



Learn / Power Platform / Power BI /

Understand star schema and the importance for Power BI

Article • 02/28/2023 • 18 minutes to read • 9 contributors

Feedback

https://learn.microsoft.com/en-us/power-bi/guidance/star-schema

“As a result, an execution engine for the Lakehouse must have a design that is flexible enough to deliver good performance on arbitrary uncurated data, and excellent performance on data following Lakehouse best practices—multidimensional clustering [19], reasonable file sizes, and appropriate data types”

https://cs.stanford.edu/~matei/papers/2022/sigmod_photon.pdf

Determine table category

A star schema organizes data into fact and dimension tables. Some tables are used for integration or staging data before moving to a fact or dimension table. As you design a table, decide whether the table data belongs in a fact, dimension, or integration table. This decision informs the appropriate table structure.

- Fact tables contain quantitative data that are commonly generated in a transactional system, and then loaded into the data warehouse. For example, a retail business generates sales transactions every day, and then loads the data into a data warehouse fact table for analvsis.

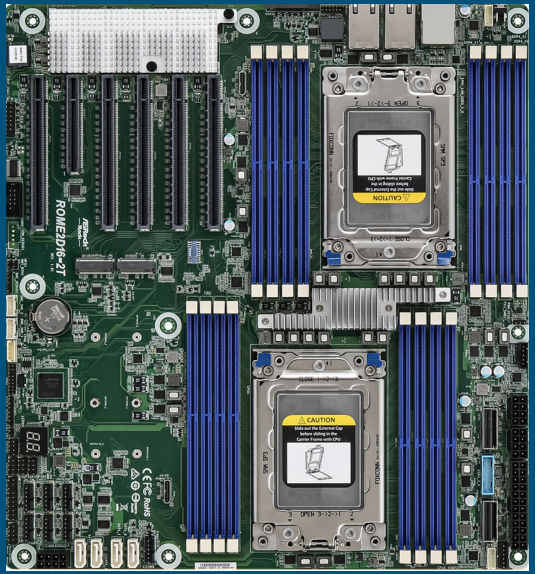
- Dimension tables contain attribute data that might change but usually changes infrequently. For example, a customer's name and address are stored in a dimension table and updated only when the customer's profile changes. To minimize the size of a large fact table, the customer's name and address don't need to be in every row of a fact table. Instead, the fact table and the dimension table can share a customer ID. A query can join the two tables to associate a customer's profile and transactions.

https://learn.microsoft.com/en-us/fabric/data-warehouse/tables

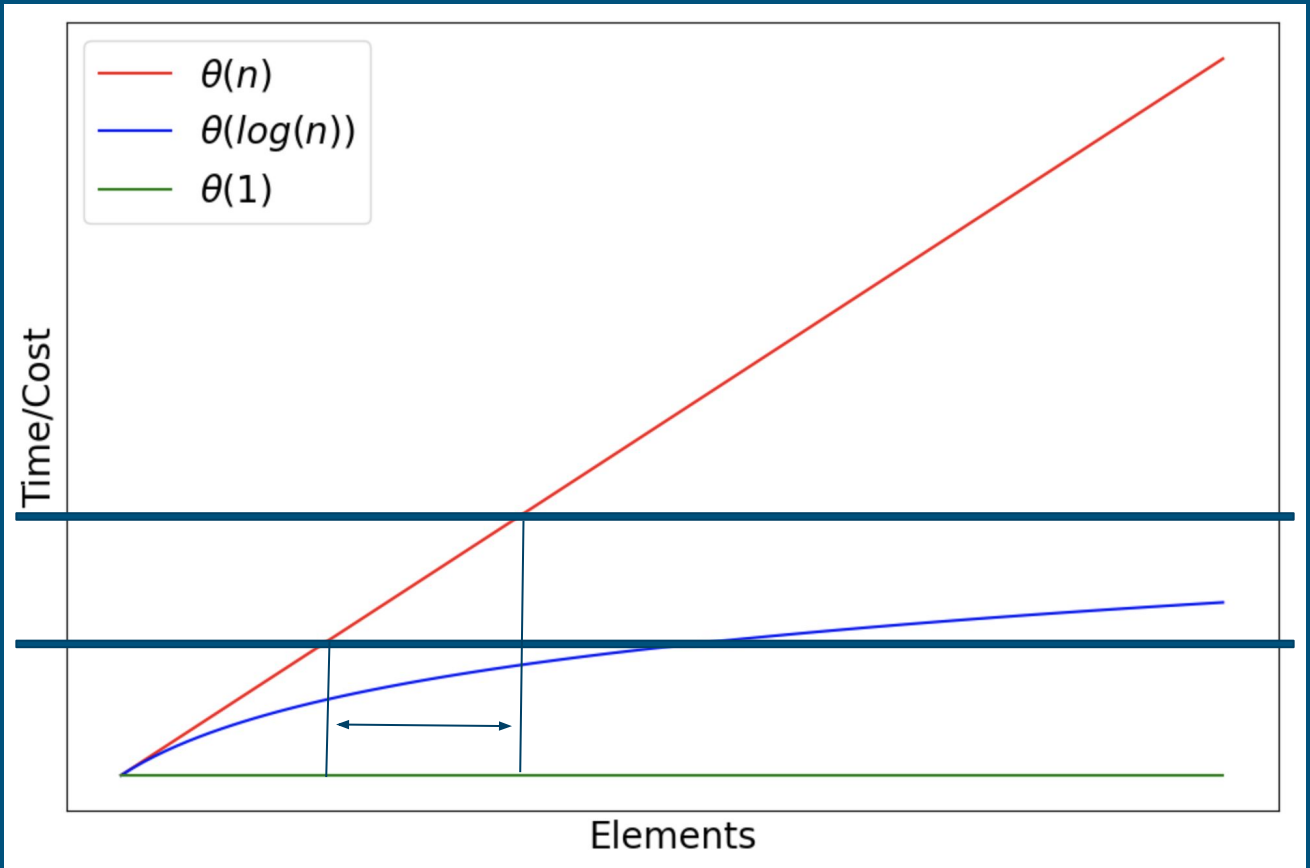
Speed of light in vacuum: 299,792,458 m/s
Speed of fiber (IOR 1.4682): 204,190,476 m/s
Speed of electric current flow in copper : about 270,000,000 m/s

Storage

Meters or kilometers



https://www.m2optics.com/blog/how-to-calculate-fiber-latency
https://www.matec-conferences.org/articles/matecconf/pdf/2019/29/matecconf_icsbe2019_05022.pdf
https://www.sciencefocus.com/science/how-fast-does-electricity-flow/


$\theta(n)$
$\theta(log(n))$
$\theta(1)$
Time/Cost
Elements

2.2 million Blog posts in an 1 core Azure SQL Database

	PostId	Title	Content	BlogId	Created	RowGuid
1	814595	Lorem ipsum dolor sit ame…	Lorem ipsum dolor sit ame…	840003	2023-03-02 20.51.19.11902…	9bb89e0b-2a07-4dcc-8ad3-b…
2	814596	Lorem ipsum dolor sit ame…	Lorem ipsum dolor sit ame…	840004	2023-03-02 20.51.19.84626…	223465c1-2481-4200-8fd8-c…
3	814597	Lorem ipsum dolor sit ame…	Lorem ipsum dolor sit ame…	840004	2023-03-02 20.51.19.84708…	d8e93c56-8cf4-459f-9284-4…
4	814598	Lorem ipsum dolor sit ame…	Lorem ipsum dolor sit ame…	840004	2023-03-02 20.51.19.84734…	329cbe03-fa0c-42ce-8c94-9…
5	814599	Lorem ipsum dolor sit ame…	Lorem ipsum dolor sit ame…	840004	2023-03-02 20.51.19.84740…	9dd6ccf7-a0ed-4635-b699-b…
6	814600	Lorem ipsum dolor sit ame…	Lorem ipsum dolor sit ame…	840004	2023-03-02 20.51.19.84744…	fa913a97-38cb-4a23-821c-3…
7	814601	Lorem ipsum dolor sit ame…	Lorem ipsum dolor sit ame…	840004	2023-03-02 20.51.19.84747…	0eaf50f4-feac-4875-a7a6-1…
8	814602	Lorem ipsum dolor sit ame…	Lorem ipsum dolor sit ame…	840004	2023-03-02 20.51.19.84750…	88a3ef46-11c2-4051-b482-5…
9	814603	Lorem ipsum dolor sit ame…	Lorem ipsum dolor sit ame…	840005	2023-03-02 20.51.19.84755…	6f5dccad-c77f-4fa4-9e2f-6…
10	814604	Lorem ipsum dolor sit ame…	Lorem ipsum dolor sit ame…	840005	2023-03-02 20.51.19.84762…	dbb01697-1f23-41df-ae14-d…

```sql
SET STATISTICS IO, TIME ON

SELECT TOP(1) * FROM [Posts]
ORDER BY [Created] DESC
```

Results | Messages

12:05:48 Started executing query at Line 1

(1 row affected)

Table 'Posts'. Scan count 1, logical reads 213899, phy
reads 0, lob page server read-ahead reads 0.

Table 'Worktable'. Scan count 0, logical reads 0, phys
0, lob page server read-ahead reads 0.

SQL Server Execution Times:

CPU time = 5453 ms, elapsed time = 24543 ms.

Total execution time: 00:00:24.588

```sql
SET STATISTICS IO, TIME ON

SELECT TOP(1) * FROM [Posts]
ORDER BY [Created] DESC
```

Results | Messages

12:07:47 Started executing query at Line 1

SQL Server parse and compile time:

CPU time = 0 ms, elapsed time = 0 ms.

SQL Server Execution Times:

CPU time = 0 ms, elapsed time = 0 ms.

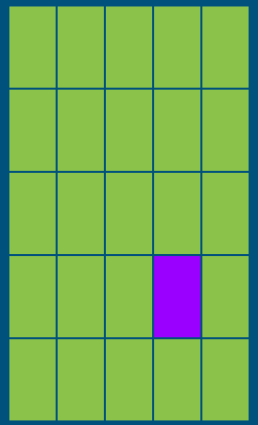
(1 row affected)

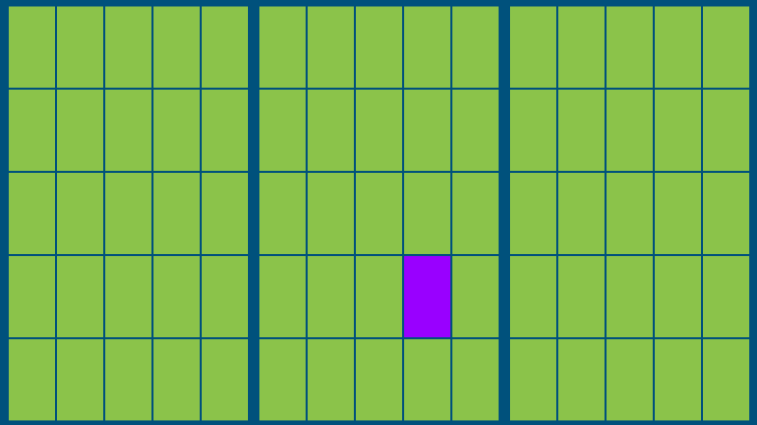
Table 'Posts'. Scan count 1, logical reads 4, phy
page server read-ahead reads 0.

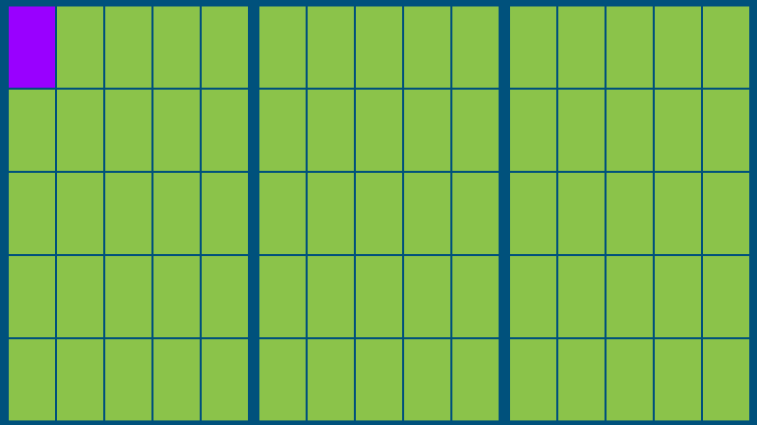
SQL Server Execution Times:

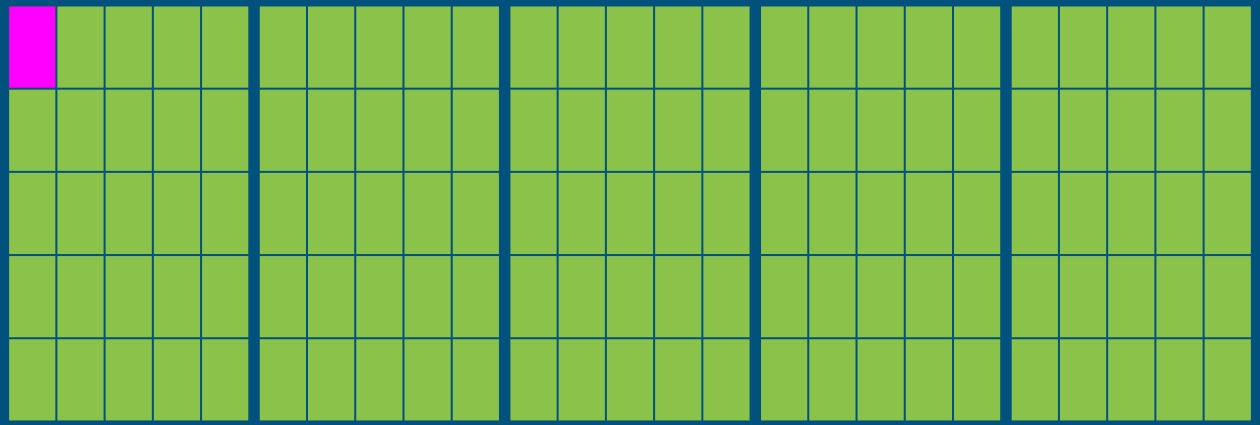
CPU time = 0 ms, elapsed time = 0 ms.

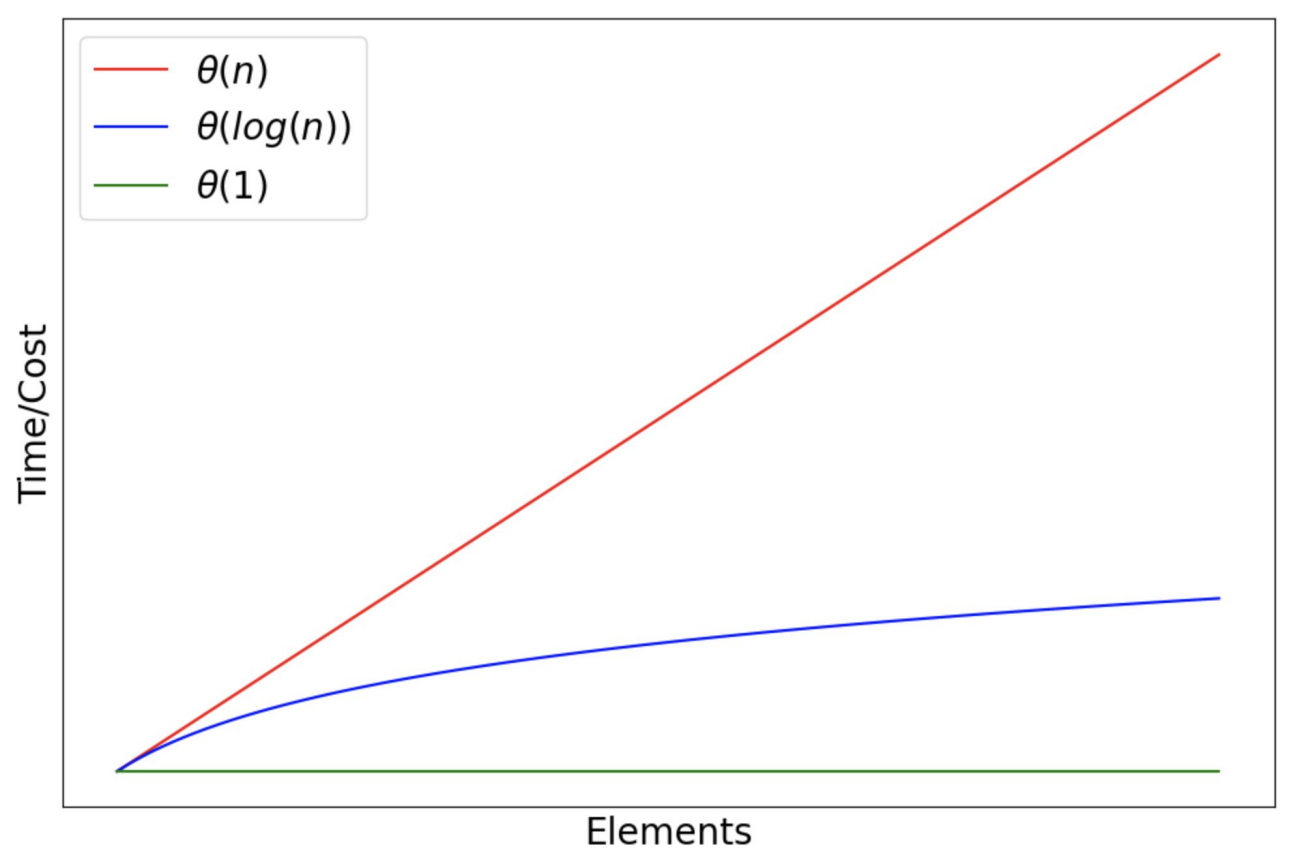
Total execution time: 00:00:00.021










$\theta(n)$
$\theta(log(n))$
$\theta(1)$
Time/Cost
Elements

QUERY COST = CODE + DATA ARRANGEMENT

- Reusable models
- End-user friendly
- Referential Consistent
- Deterministic
- Performant
- Scalable

- Reusable models
- End-user friendly
- Referential Consistent
- Deterministic
- Performant
- Scalable

```
SELECT DATEPART(m, Date) AS Month, AVG(Units) FROM Usages
WHERE Date BETWEEN ‘2022-01-01’ AND ‘2022-12-31’
AND ClientId != ‘06d86970-4a85-40b3-9726-440c7123d2eb’
GROUP BY DATEPART(m, Date)
```

A) 2022-01-01 is included and 2022-01-31 is excluded

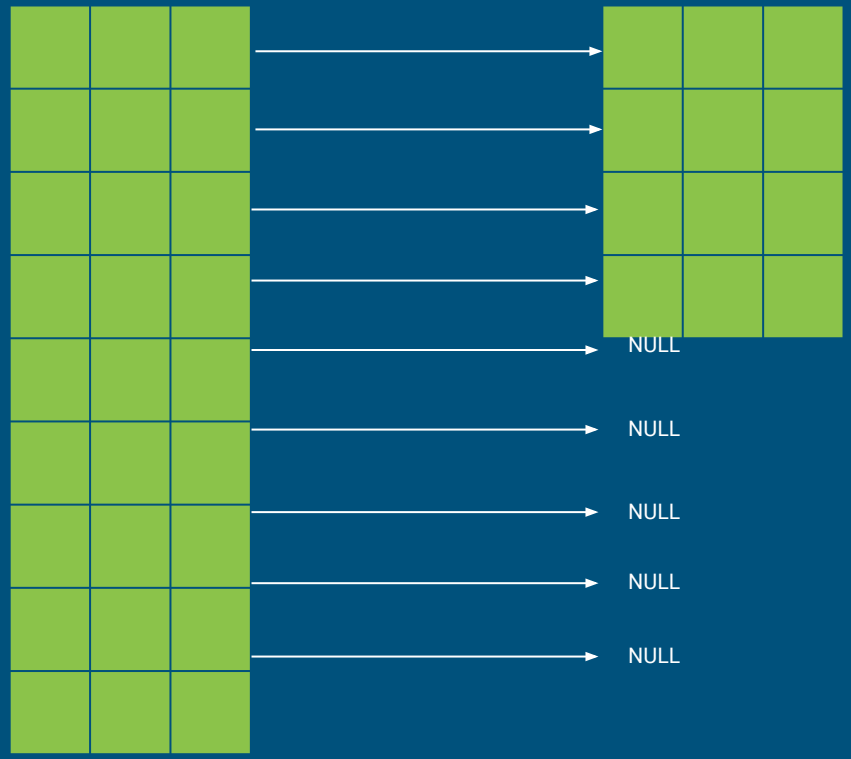
B) 2022-01-01 is excluded and 2022-01-31 is included

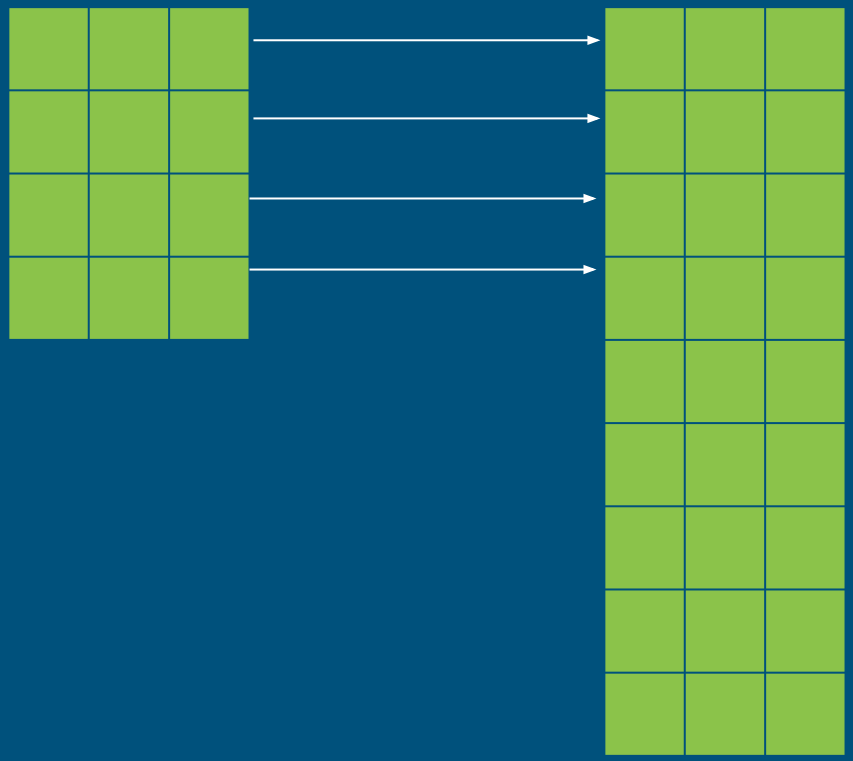
C) 2022-01-01 is included and 2022-01-31 is included

D) 2022-01-01 is excluded and 2022-01-31 is excluded

```
SELECT MonthName AS Month, AVG(Units) FROM Fact.Usages
INNER JOIN dim.Dates ON Fact.Usages.DateDWHId =
dim.Dates.DWHId
WHERE dim.Dates.year = 2022
GROUP BY MonthName
```

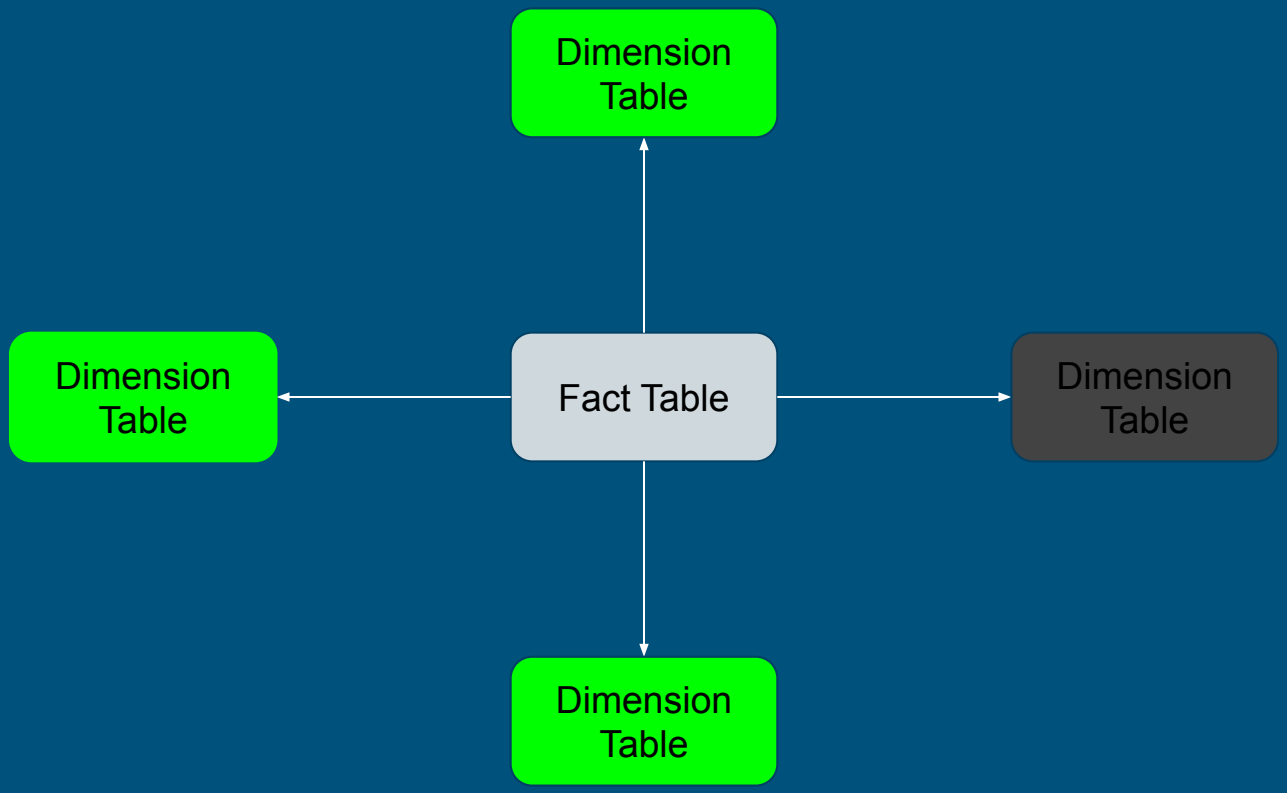
- Reusable models
- End-user friendly
- Referential Consistent
- Deterministic
- Performant
- Scalable

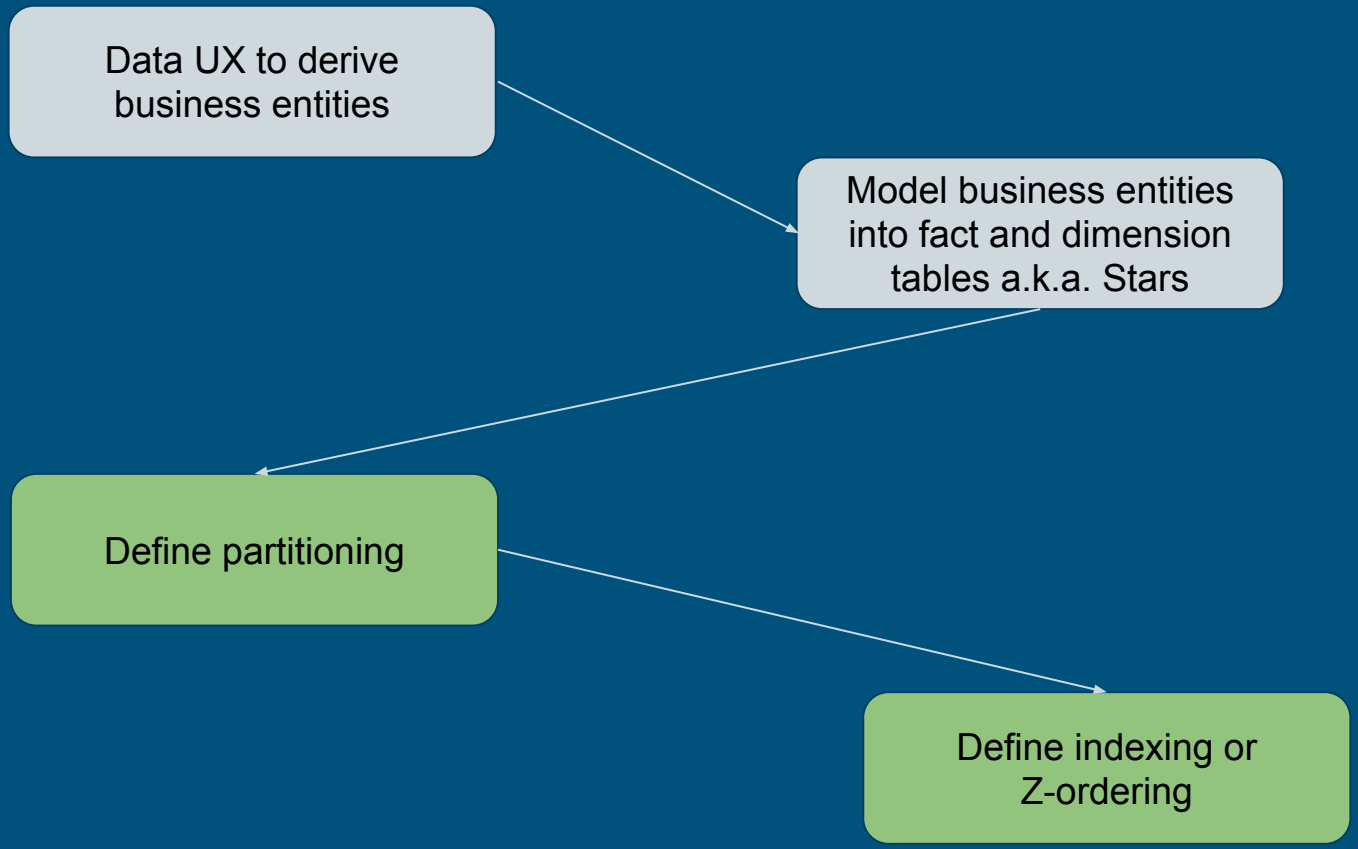

NULL
NULL
NULL
NULL
NULL



- Reusable models
- End-user friendly
- Referential Consistent
- Deterministic
- Performant
- Scalable

- Reusable models
- End-user friendly
- Referential Consistent
- Deterministic
- Performant
- Scalable


Dimension Table
Dimension Table
Fact Table
Dimension Table
Dimension Table


Data UX to derive business entities
Model business entities into fact and dimension tables a.k.a. Stars
Define partitioning
Define indexing or Z-ordering

BUSINESS LANGUAGE

VS

TECH LANGUAGE

Our Imaginary Company

- Scales from 0 to brutto 100.000 customers over a period of 5 whole years (2018-2022)
- They gained on average 50 new customers per day
- They experience a 15% churn per day
- They have B2C, B2B subscription. Employees are B2B customers with a free subscription
- The sales are consumption based
- 80% of the customers on a day, use their service once per day
- 20% of the customers on a day, use their service 2 times per day
- 10% of the customers on a day, use their service 3 times per day

Customers

B2B, B2C categories

Active customers

Has 1+ Subscription

Customer’s address is Customer’s location

CRM Id 9999 is a test customer

Subscriptions

B2B, B2C categories

Has Name

Has 1 product

Product can change

Active subscriptions, can have notice date, until they are churned

Products

B2B, B2C categories

Has Name

Has price

Price is adjusted once a year

Consumption

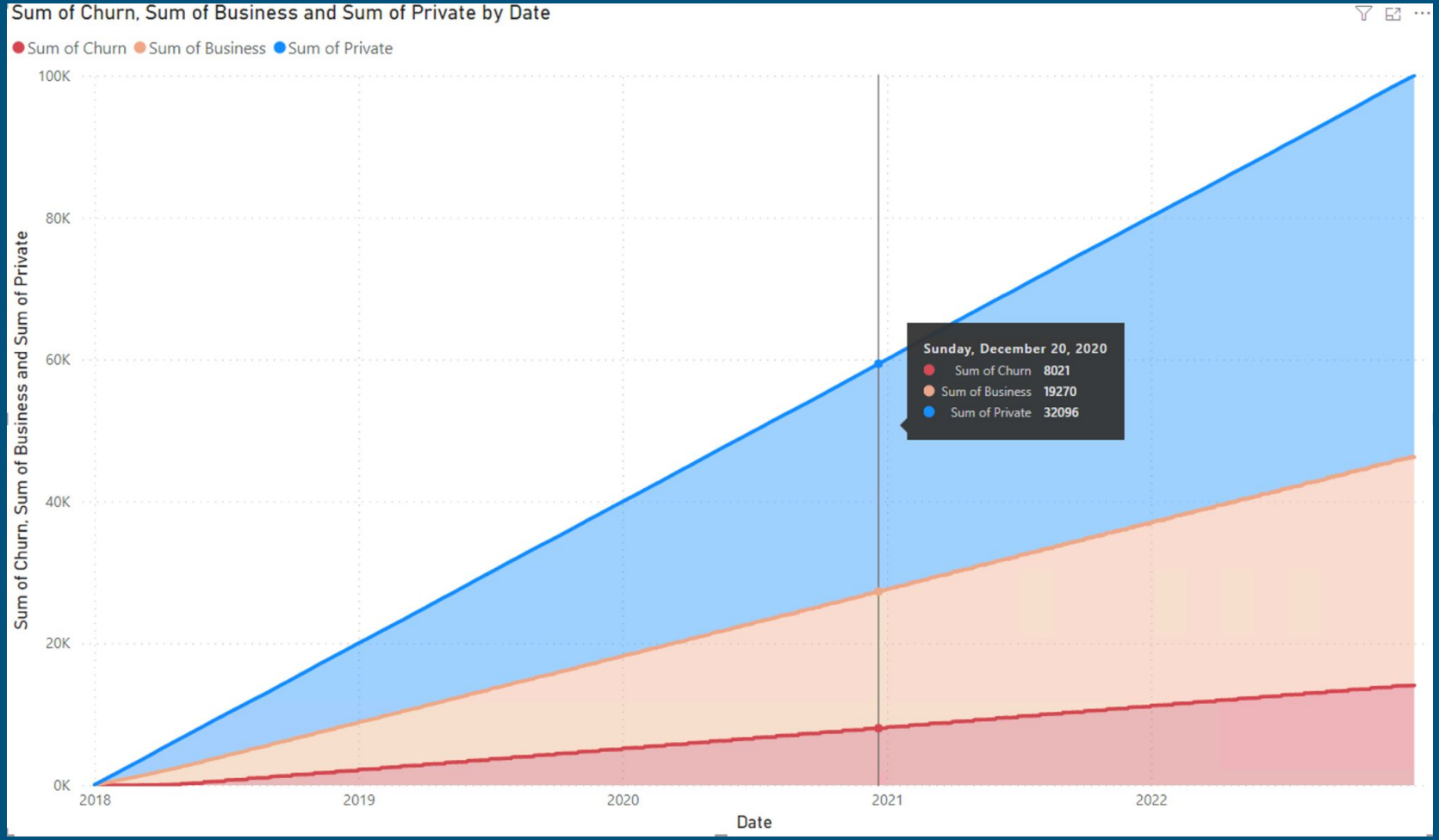
Customer do some consumption

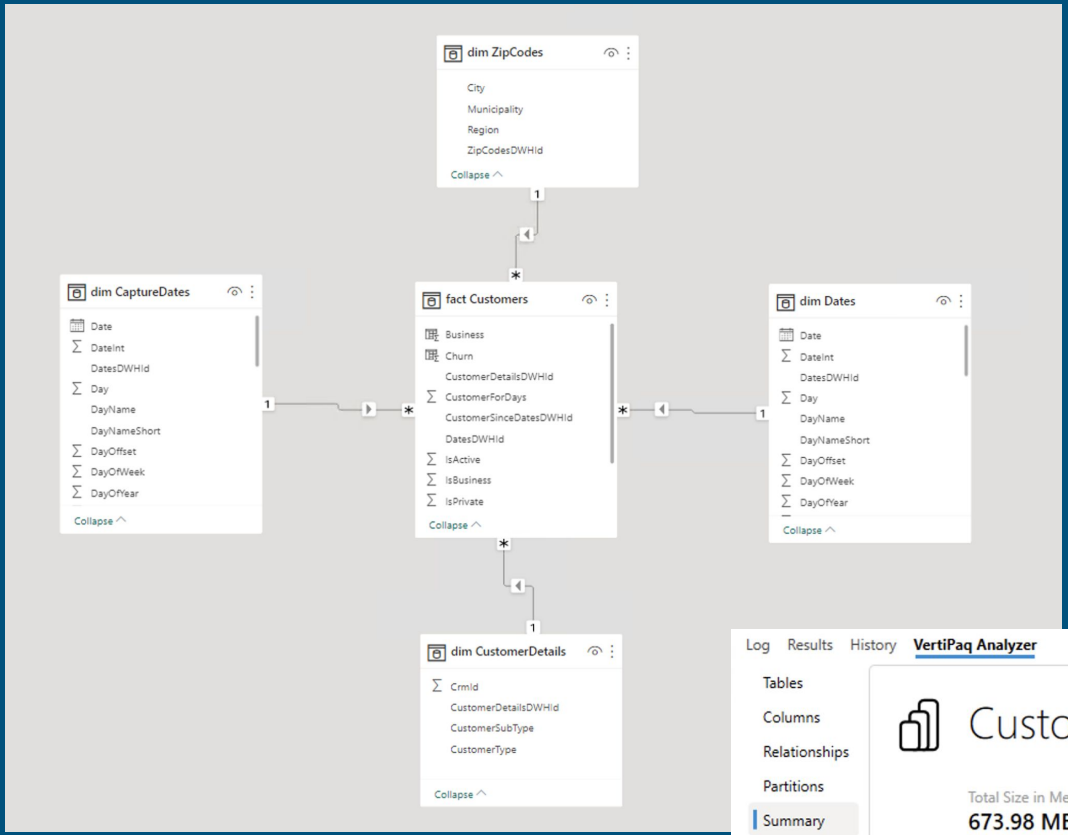
Usage most be tied to a subscription, to decide the price

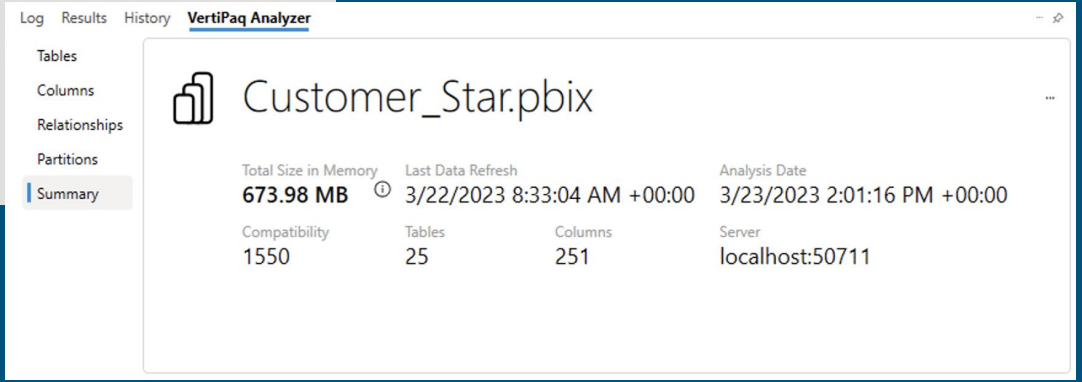
Usage is done at some location

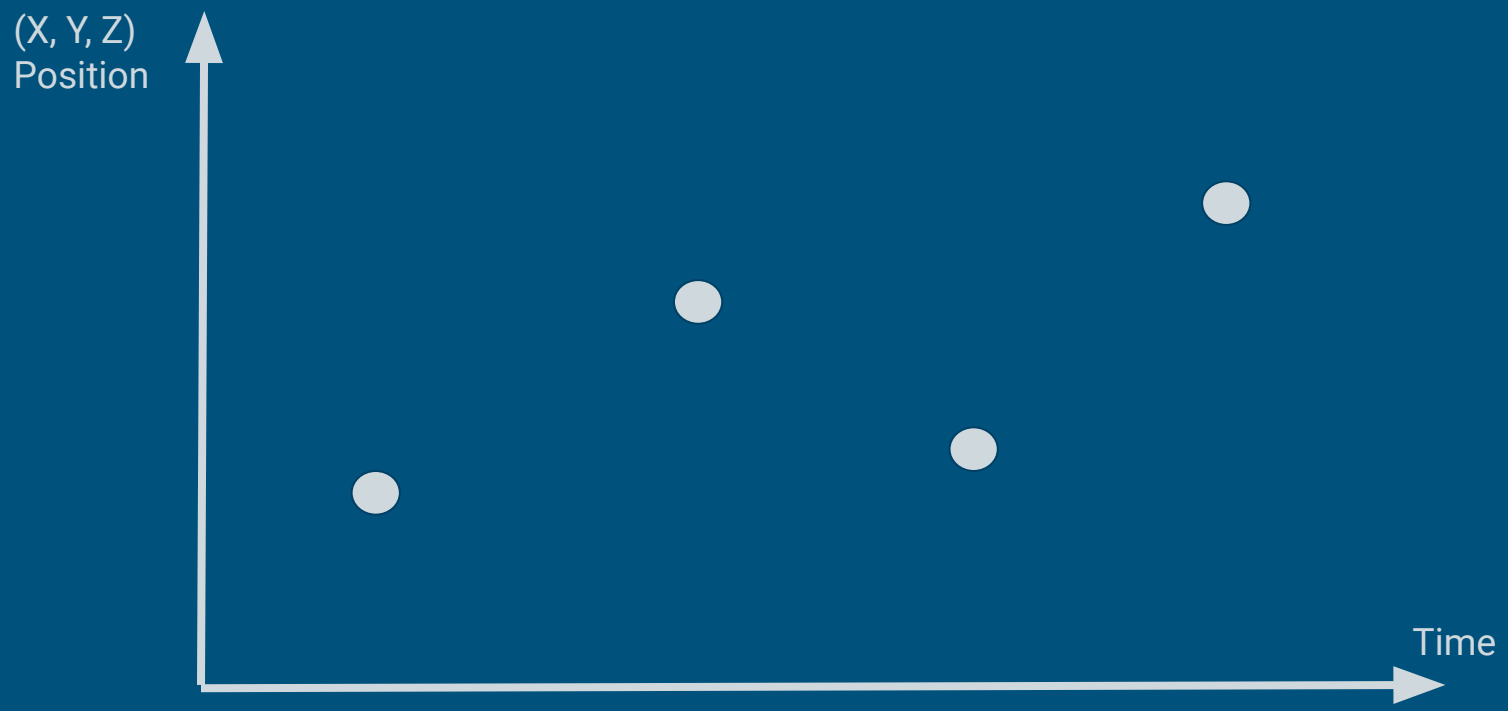
Our Imaginary Company DWH in Azure SQL Database

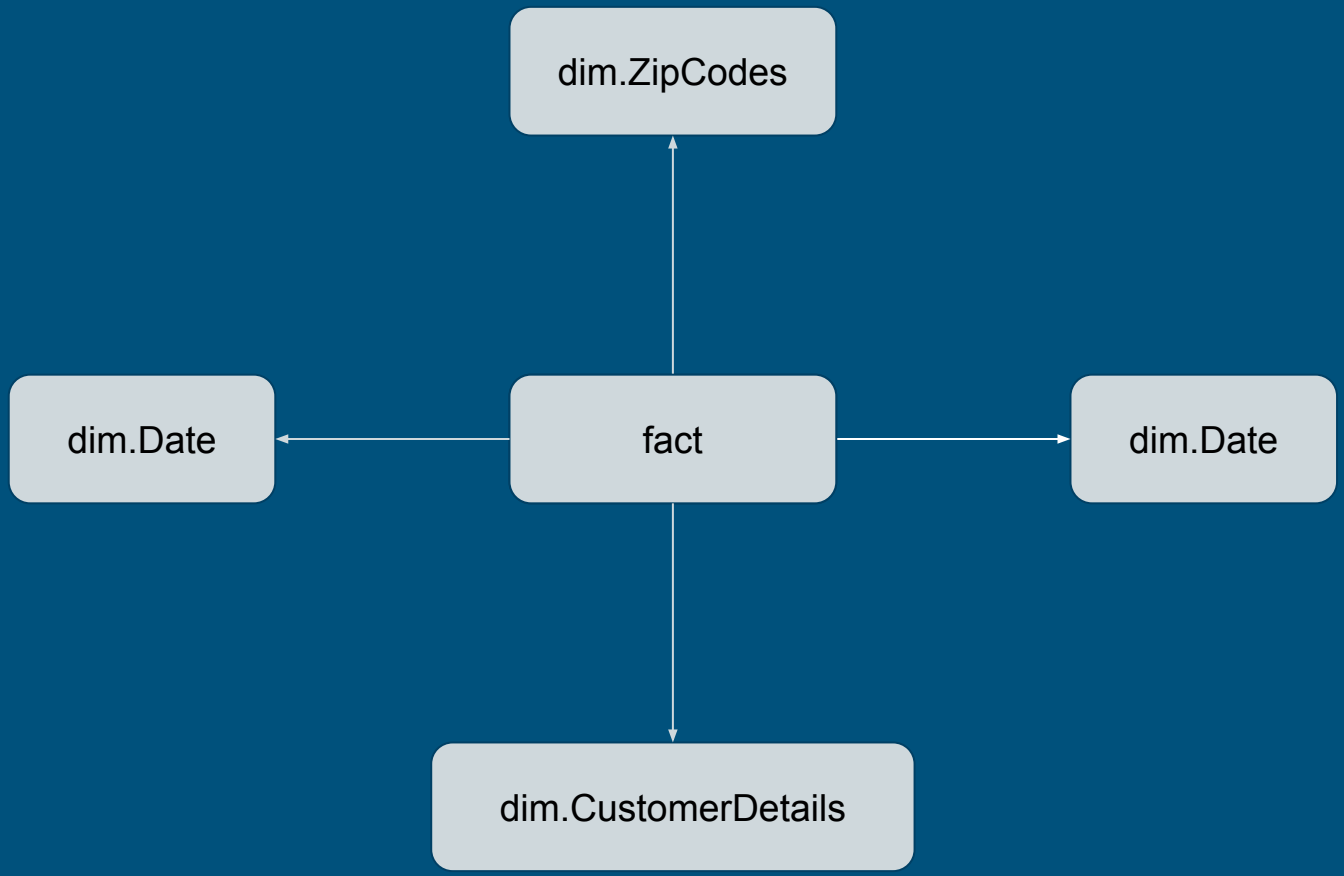
TableName	NumberOfRows
fact.Customers	91,296,503
fact.Subscriptions	91,296,503
fact.Consumptions	86,736,525
dim.CustomerDetails	100,000
dim.Times	86,401
dim.Dates	2,557
dim.ZipCodes	1,122
dim.ProductDetails	15
dim.SubscriptionDetails	4
Total	**269,519,630**


Sum of Churn, Sum of Business and Sum of Private by Date
Sum of Churn
Sum of Business
Sum of Private
100K
80K
60K
40K
20K
0K
Sum of Churn, Sum of Business and Sum of Private
2018
2019
2020
2021
2022
Date
Sunday, December 20, 2020
Sum of Churn 8021
Sum of Business 19270
Sum of Private 32096


dim ZipCodes
City
Municipality
Region
ZipCodesDWHId
Collapse
dim CaptureDates
Date
DateInt
DatesDWHId
Day
DayName
DayNameShort
DayOffset
DayOfWeek
DayOfYear
Collapse
fact Customers
Business
Churn
CustomerDetailsDWHId
CustomerForDays
CustomerSinceDatesDWHId
DatesDWHId
IsActive
IsBusiness
IsPrivate
Collapse
dim Dates
Date
DateInt
DatesDWHId
Day
DayName
DayNameShort
DayOffset
DayOfWeek
DayOfYear
Collapse
dim CustomerDetails
CrmId
CustomerDetailsDWHId
CustomerSubType
CustomerType
Collapse


Log
Results
History
VertiPaq Analyzer
Tables
Columns
Relationships
Partitions
Summary
Customer_Star.pbix
Total Size in Memory
673.98 MB
Last Data Refresh
3/22/2023 8:33:04 AM +00:00
Analysis Date
3/23/2023 2:01:16 PM +00:00
Compatibility
1550
Tables
25
Columns
251
Server
localhost:50711


(X, Y, Z)
Position
Time


dim.ZipCodes
dim.Date
fact
dim.Date
dim.CustomerDetails

fact.Customers

	DatesDWHId	CustomerDetailsDWHId	CustomerSinceDatesDWHId	ZipCodesDWHId	CustomerForDays	NumberOfSubcriptions	IsActive	IsBusiness	IsPrivate
1	20200821	15193	20181006	1667	19815	1	1	0	1
2	20221109	58477	20201205	8660	19904	1	1	1	0
3	20200425	40165	20200105	5970	320	1	1	1	0
4	20211203	39078	20191216	1363	19987	1	1	0	1
5	20200713	793	20180115	1092	313	0	0	0	1
6	20220228	35914	20191020	4880	29208	1	1	1	0
7	20221101	9242	20180618	1318	495	0	0	1	0
8	20220222	67412	20210514	1823	211	0	0	0	1
9	20200831	152	20180103	1366	20728	1	1	1	0
10	20191030	25130	20190406	8250	624	1	1	1	0

- dbo.expensive_types
  - Columns
    - As_INT32 (int, null)
    - As_VARCHAR (varchar(11), null)
    - As_NVARCHAR (nvarchar(11), null)
    - As_GUID (uniqueidentifier, null)
    - As_DATETIME2 (datetime2(7), null)

	As_INT32	As_VARCHAR	As_NVARCHAR	As_GUID	As_DATETIME2
1	1	1	1	5A04007B-9F04-4A0C-9F55-ECF4FAC035D8	2020-01-01 00:00:01.0000000
2	2	2	2	CABB555C-5FE4-4011-AEC8-A4A5F94B2925	2020-01-01 00:00:02.0000000
3	3	3	3	E25E2AAD-0D31-4449-9C82-2592E8B42F5E	2020-01-01 00:00:03.0000000
4	4	4	4	EDFAD413-D2CC-4932-A6C1-CA4041D9C3CE	2020-01-01 00:00:04.0000000
5	5	5	5	168D5FE0-CED5-4963-9360-F6721BABF014	2020-01-01 00:00:05.0000000
6	6	6	6	7610CF70-12C3-430C-9491-F6CAB8DA1557	2020-01-01 00:00:06.0000000

Log Results History **VertiPaq Analyzer**

Tables
Columns
Relationships
Partitions
Summary

Name	Cardinality	Total Size	Data	Dictionary	Hier Size	Encoding	Data Type
expensive_types	**1,000,000**	**271,271,004**	**13,334,128**	**220,793,940**	**36,000,072**	**Many**	-
As_GUID	1,000,000	110,236,648	2,666,800	99,569,832	8,000,016	HASH	String
As_DATETIME2	1,000,000	52,704,636	2,666,800	42,037,820	8,000,016	HASH	DateTime
As_VARCHAR	1,000,000	50,259,824	2,666,800	39,593,008	8,000,016	HASH	String
As_NVARCHAR	1,000,000	50,259,824	2,666,800	39,593,008	8,000,016	HASH	String
As_INT32	1,000,000	6,666,944	2,666,800	136	4,000,008	VALUE	Int64

expensive_types.pbix

Total Size in Memory	Last Data Refresh	Analysis Date
258.80 MB	3/23/2023 10:32:38 AM +00:00	3/23/2023 10:32:48 AM +00:00

Compatibility	Tables	Columns	Server
1550	3	22	localhost:49911

dim.CustomerDetails

	CustomerDetailsDWHId	CrmId	CustomerType	CustomerSubType
1	1	1	Private	Regular
2	2	6	Business	Employee
3	3	11	Private	Regular
4	4	16	Private	Regular
5	5	21	Business	Regular
6	6	26	Private	Regular

dim.ZipCodes

ZipCodesDWHId	City	Municipality	Region
4070	Kirke Hyllinge	Lejre Kommune	Region Sjælland
4100	Ringsted	Næstved Kommune	Region Sjælland
4130	Viby Sjælland	Roskilde Kommune	Region Sjælland
4140	Borup	Ringsted Kommune	Region Sjælland
4160	Herlufmagle	Næstved Kommune	Region Sjælland
4171	Glumsø	Næstved Kommune	Region Sjælland

dim.Dates

	DatesDWHId	Date	Year	StartOfYear	EndOfYear	YearQuarter	YearMonth	YearWeek	Month	StartOfMonth	EndOfMonth	DaysInMonth	Day	DayName	DayNameShort
787	20200225	2020-02-25	2020	2020-01-01	2020-12-31	2020-Q1	2020-02	2020-9	2	2020-02-01	2020-02-29	29	25	Tuesday	Tue
788	20200226	2020-02-26	2020	2020-01-01	2020-12-31	2020-Q1	2020-02	2020-9	2	2020-02-01	2020-02-29	29	26	Wednesday	Wed
789	20200227	2020-02-27	2020	2020-01-01	2020-12-31	2020-Q1	2020-02	2020-9	2	2020-02-01	2020-02-29	29	27	Thursday	Thu
790	20200228	2020-02-28	2020	2020-01-01	2020-12-31	2020-Q1	2020-02	2020-9	2	2020-02-01	2020-02-29	29	28	Friday	Fri
791	20200229	2020-02-29	2020	2020-01-01	2020-12-31	2020-Q1	2020-02	2020-9	2	2020-02-01	2020-02-29	29	29	Saturday	Sat
792	20200301	2020-03-01	2020	2020-01-01	2020-12-31	2020-Q1	2020-03	2020-9	3	2020-03-01	2020-03-31	31	1	Sunday	Sun

```sql
SET STATISTICS IO, TIME ON

SELECT City, SUM([IsActive]) Active, SUM([IsBusiness]) Business, SUM([IsPrivate]) Private
   FROM [fact].[Customers]
   INNER JOIN [dim].[ZipCodes] ON [dim].[ZipCodes].[ZipCodesDWHId] = [fact].[Customers].[ZipCodesDWHId]
   WHERE IsActive = 1
   AND DatesDWHId = '20221231'
   GROUP BY City
   Order by Active DESC
```

100 %

Results | Messages

	City	Active	Business	Private
1	København K	18426	6921	11505
2	København V	12577	4721	7856
3	Frederiksberg C	8193	3101	5092
4	Fredericia	157	52	105
5	Thorsø	90	36	54
6	Snertinge	88	35	53
7	Nordborg	88	34	54
8	Spøttrup	87	37	50
9	Maribo	86	32	54
10	Langebæk	86	32	54

```sql
SET STATISTICS IO, TIME ON

SELECT City, SUM([IsActive]) Active, SUM([IsBusiness]) Business, SUM([IsPrivate]) Private
   FROM [fact].[Customers]
   INNER JOIN [dim].[ZipCodes] ON [dim].[ZipCodes].[ZipCodesDWHId] = [fact].[Customers].[ZipCodesDWHId]
   WHERE IsActive = 1
   AND DatesDWHId = '20221231'
   GROUP BY City
   Order by Active DESC
```

100 %

Results | Messages

```
SQL Server parse and compile time:
   CPU time = 0 ms, elapsed time = 7 ms.

(607 rows affected)
Table 'Customers'. Scan count 1, logical reads 0, physical reads 0, page server reads 0, read-ahead reads 0, pag
Table 'Customers'. Segment reads 2, segment skipped 86.
Table 'ZipCodes'. Scan count 1, logical reads 24, physical reads 0, page server reads 0, read-ahead reads 0, pag
Table 'Worktable'. Scan count 0, logical reads 0, physical reads 0, page server reads 0, read-ahead reads 0, pag
Table 'Worktable'. Scan count 0, logical reads 0, physical reads 0, page server reads 0, read-ahead reads 0, pag

 SQL Server Execution Times:
   CPU time = 0 ms,  elapsed time = 5 ms.

Completion time: 2023-03-22T16:40:11.4233847+00:00
```

100 %

```sql
SET STATISTICS IO, TIME ON

SELECT City, SUM([IsActive]) Active, SUM([IsBusiness]) Business, SUM([IsPrivate]) Private
  FROM [fact].[Customers]
  INNER JOIN [dim].[ZipCodes] ON [dim].[ZipCodes].[ZipCodesDWHId] = [fact].[Customers].[ZipCodesDWHId]
  WHERE IsActive = 1
  AND DatesDWHId = '20191231'
  GROUP BY City
  Order by Active DESC
```

100 %

Results | Messages

```
 SQL Server Execution Times:
   CPU time = 0 ms,  elapsed time = 0 ms.

(607 rows affected)
Table 'Customers'. Scan count 1, logical reads 0, physical reads 0, page server reads 0, read-ahead reads 0, pag
Table 'Customers'. Segment reads 16, segment skipped 72.
Table 'ZipCodes'. Scan count 1, logical reads 24, physical reads 0, page server reads 0, read-ahead reads 0, pag
Table 'Worktable'. Scan count 0, logical reads 0, physical reads 0, page server reads 0, read-ahead reads 0, pag
Table 'Worktable'. Scan count 0, logical reads 0, physical reads 0, page server reads 0, read-ahead reads 0, pag

 SQL Server Execution Times:
   CPU time = 15 ms,  elapsed time = 8 ms.
```

100 %

```sql
SET STATISTICS TIME, IO ON

SELECT [Month], [MonthName], Sum([Usage]) AS Usage, Sum([Cost]) AS Cost
   FROM [fact].[Consumptions]
   INNER JOIN dim.Dates ON dim.Dates.DatesDWHId = [fact].[Consumptions].DatesDWHId
   WHERE dim.Dates.Year = 2021
   GROUP BY [Month], [MonthName]
   ORDER BY [Month]
```

100 %

Results | Messages

	Month	MonthName	Usage	Cost
1	1	January	884983.340000005	108416284.200002
2	2	February	820100.75999999	100491573.000002
3	3	March	932301.769999989	114237664.799999
4	4	April	924773.170000015	113288848.399997
5	5	May	979593.96000001	119957490.800002
6	6	June	971217.450000006	118973220.800002
7	7	July	1027025.87000002	125811991.2
8	8	August	1052319.40999999	128934976.799999
9	9	September	1041859.39000001	127611968.4
10	10	October	1100325.65000003	134791001.800002
11	11	November	1088203.36999999	133286483.399999
12	12	December	1147489.87	140599759.999999

SQLQuery2.sql - dw...icrosoft.com (86))* | SQLQuery1.sql - dw...icrosoft.com (54)) | ~vs41F2.sql - dwhs...micros

```sql
SET STATISTICS TIME, IO ON

SELECT [Month], [MonthName], Sum([Usage]) AS Usage, Sum([Cost]) AS Cost
   FROM [fact].[Consumptions]
   INNER JOIN dim.Dates ON dim.Dates.DatesDWHId = [fact].[Consumptions].DatesDWHId
   WHERE dim.Dates.Year = 2021
   GROUP BY [Month], [MonthName]
   ORDER BY [Month]
```

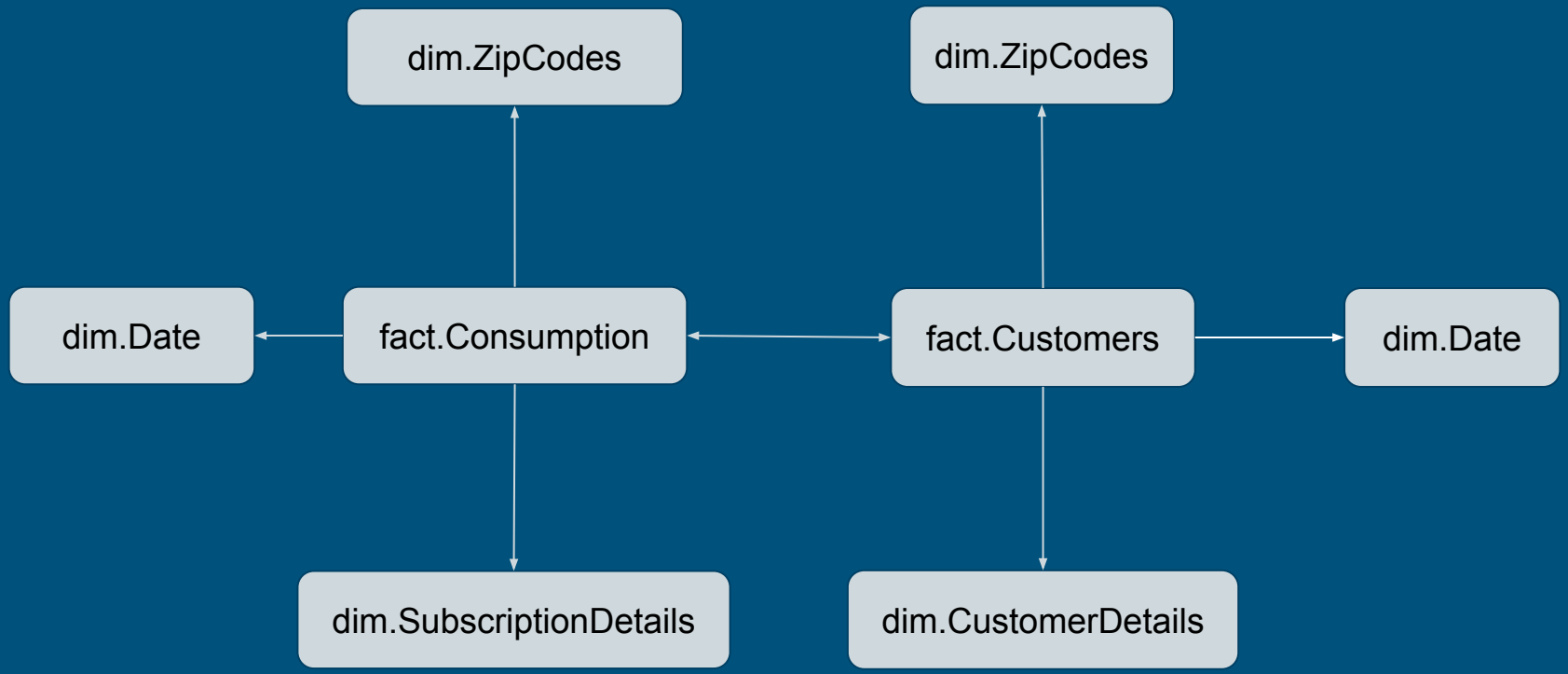
100 %

Results | Messages

```
SQL Server parse and compile time:
   CPU time = 6 ms, elapsed time = 6 ms.

(12 rows affected)
Table 'Consumptions'. Scan count 1, logical reads 0, physical reads 0, page server reads 0,
Table 'Consumptions'. Segment reads 24, segment skipped 59.
Table 'Dates'. Scan count 1, logical reads 66, physical reads 0, page server reads 0, read-a
Table 'Worktable'. Scan count 0, logical reads 0, physical reads 0, page server reads 0, rea

 SQL Server Execution Times:
   CPU time = 735 ms,  elapsed time = 745 ms.
```



https://www.kimballgroup.com/data-warehouse-business-intelligence-resources/kimball-techniques/dimensional-modeling-techniques/drilling-across/

Log Results History **VertiPaq Analyzer**

- Tables
- Columns
- Relationships
- Partitions
- Summary

Name	Cardinality
DateTableTemplat...	1
dim CustomerDet...	100,000
dim Dates	2,557
dim StartDates	2,557
dim ZipCodes	1,122
fact Customers	91,296,503

Customer_Star.pbix

Total Size in Memory	Last Data Refresh	Analysis Date
673.98 MB	3/22/2023 8:33:04 AM +00:00	3/23/2023 3:50:19 PM +00:00

Compatibility	Tables	Columns	Server
1550	25	251	localhost:53154

Log Results History **VertiPaq Analyzer**

- Tables
- Columns
- Relationships
- Partitions
- Summary

Table / Partition	Rows
test_maxload	40,000,000
LocalDateTable_e8e7aae5-29b6-4e03-bb94-4ce...	1,826
LocalDateTable_988af9af-46dd-4e6a-9094-562...	1,095
LocalDateTable_4842f3b3-2ce7-4989-af1c-693b...	1,095
LocalDateTable_f1b8df05-94b1-455b-a893-53ff...	730
LocalDateTable_9625eda0-1c45-4ded-b7ac-7cc...	730
LocalDateTable_daa60240-07b7-4797-b0b3-291...	730
LocalDateTable_ae47f642-b921-486b-9d23-d96...	730
LocalDateTable_603fbe22-39c6-4961-a0fe-717b...	730
LocalDateTable_712a39f1-c89d-46fa-ad27-1052...	730
LocalDateTable_70e10218-2c64-43c1-95a8-d40...	730

Customer_star_flatten.pbix

Total Size in Memory	Last Data Refresh	Analysis Date
671.90 MB	3/22/2023 7:41:42 AM +00:00	3/23/2023 3:47:02 PM +00:00

Compatibility	Tables	Columns	Server
1550	21	247	localhost:53064

Most common Column stores now a days

In-memory column store engines:

- Vertipaq
  - Power BI
  - Excel
  - (Azure) Analysis Service Tabular
  - SQL
- Databricks Photon

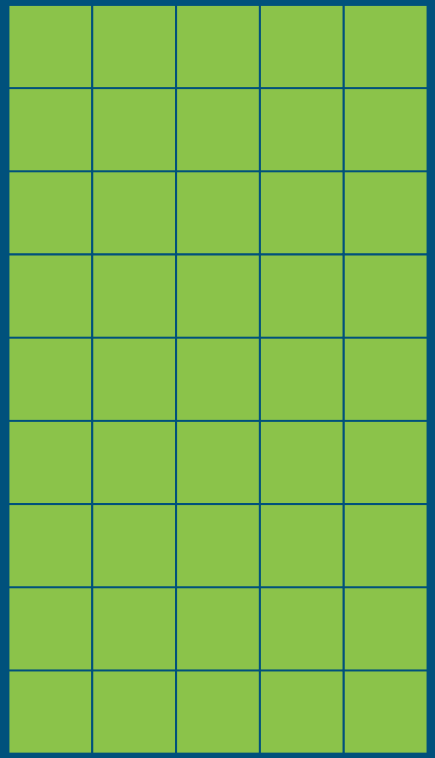
Column store storage/exchange formats

- Parquet
  - Disk storage optimized
- Arrow
  - Memory storage optimized

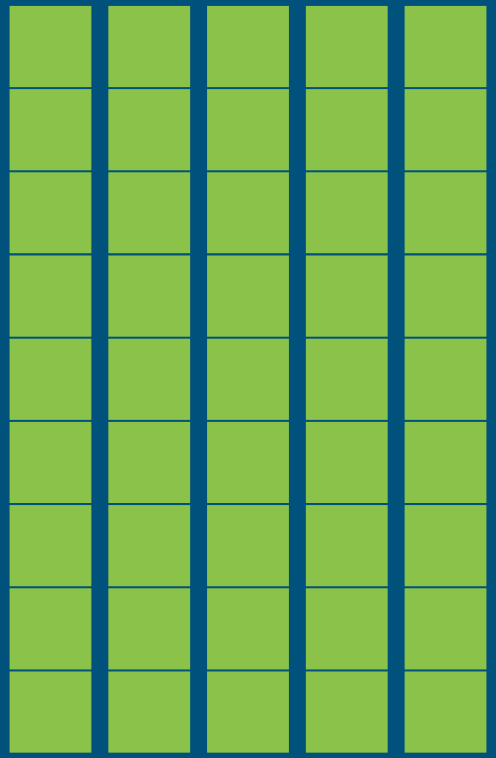
Common traits for column stores

Row groups
Integer encoding
Dictionary encoding
Run Length Encoding (RLE)
Compression of pages

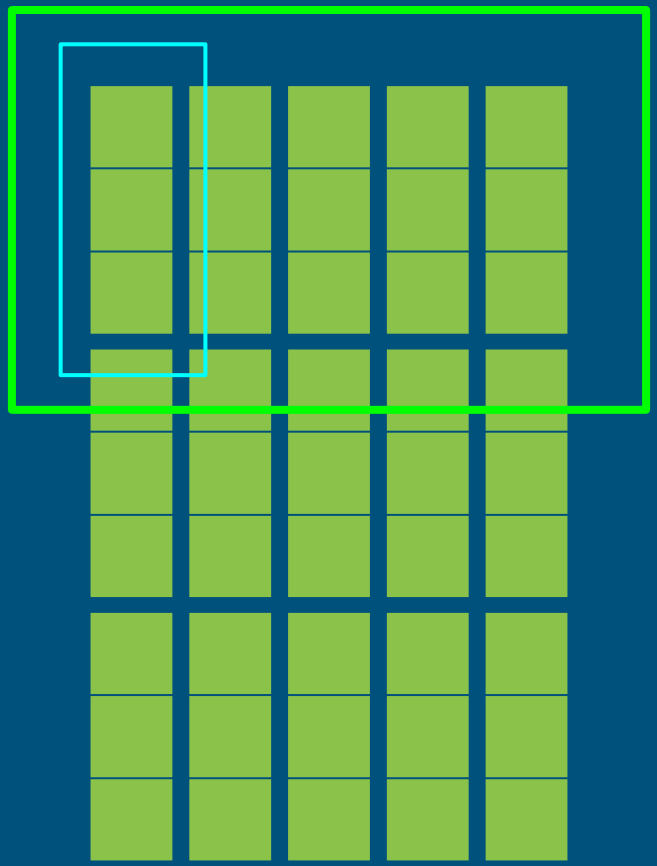
https://parquet.apache.org/docs/file-format/data-pages/encodings/
https://www.microsoftpressstore.com/articles/article.aspx?p=2449192&seqNum=3



SELECT Col1, Col2, Col3 From ….



CustomerDetailsDWHId
15193
58477
40165
39078
793
35914
9242
67412
152
25130



Direct Lake

400
500
…
…

= 0000000 0000000 00000001 10010000 = 0x00000190

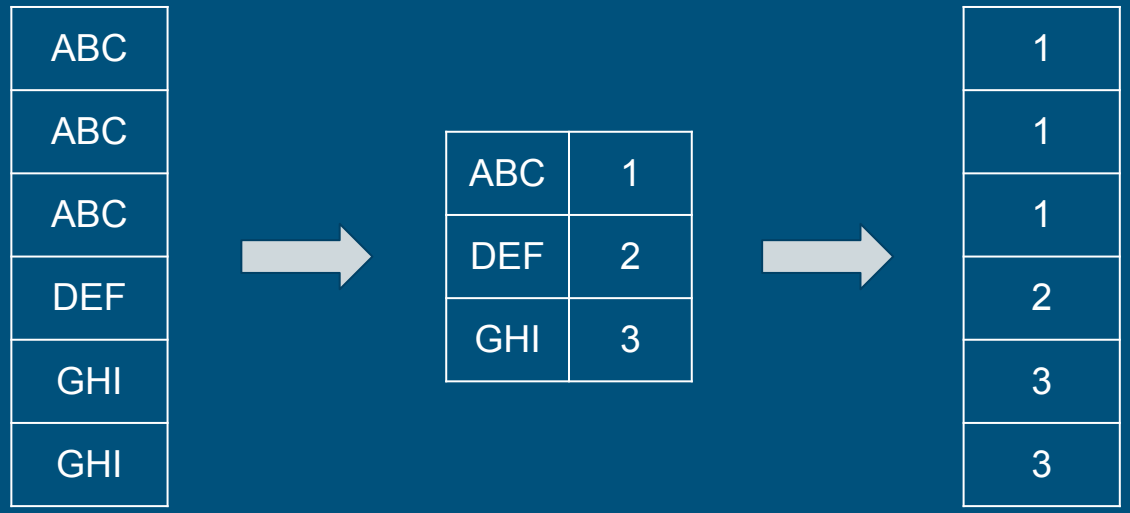
= 0000000 0000000 00000001 11111000 = 0x000001F8

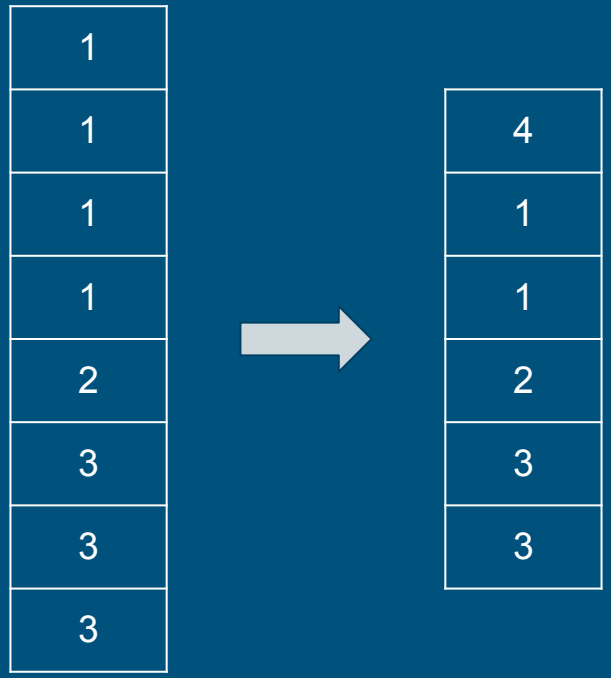
Squeezed = 00000001 10010000 00000001 11111000 = 0x019001F8

LEB128(400)= 0000011 0010000 => 00000011 10010000 = 0x00000390

LEB128(500)= 0000011 1111000 => 00000011 11111000 = 0x000003F8

Squeezed = 00000011 10010000 00000011 11111000 = 0x039003F8


ABC
ABC
ABC
DEF
GHI
GHI
ABC 1
DEF 2
GHI 3
1
1
1
2
3
3


1
1
1
1
2
3
3
3
4
1
1
2
3
3

Take aways

- Tested patterns and modelling, make us use our hardware and services better
- They can help us making more user friendly analytics platforms
- They are not that hard to implement.

Questions

ChristianHenrikReich@gmail.com